IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 27, 2017, accepted August 24, 2017, date of publication August 29, 2017, date of current version September 27, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2746095

Internal Transfer Learning for Improving Performance in Human Action Recognition for Small Datasets

TIAN WANG[1], (Member, IEEE), YANG CHEN[1], MENGYI ZHANG[2], JIE CHEN[3], (Member, IEEE), AND HICHEM SNOUSSI[4]

[1]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[2]College of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing 211800, China
[3]Centre of Intelligent Acoustics and Immersive Communications, School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
[4]Institute Charles Delaunay-LM2S-UMR STMR 6279 CNRS, University of Technology of Troyes, 10010 Troyes, France

Corresponding authors: Tian Wang (wangtian@buaa.edu.cn) and Mengyi Zhang (myzhang@njtech.edu.cn)

The work of T. Wang was supported in part by NSFC under Grant U1435220 and Grant 61503017, in part by the Fundamental Research Funds for the Central Universities under Grant YWF-14-RSC-102, and in part by the Aeronautical Science Foundation of China under Grant 2016ZC51022. The work of J. Chen was supported by NSFC under Grant 61671382. The work of H. Snoussi was supported by ANR AutoFerm Project and the Platform CAPSEC funded by Région Champagne-Ardenne and FEDER.

ABSTRACT Human action recognition nowadays plays a key role in varieties of computer vision applications. Many computer vision methods focus on algorithms designing classifiers with handcrafted features which are complex and inflexible. In this paper, we focus on the human action recognition problem and utilize 3D convolutional neural networks to automatically extract both spatial and temporal features for classification. Specifically, in order to address the training problems with small data sets, we propose an internal transfer learning strategy adapted to this framework, by incorporating the sub-data classification method into transfer learning. We evaluate our method on several data sets and obtain promising results. With the proposed strategy, the performance of human action recognition is improved obviously.

INDEX TERMS Action recognition, 3D convolutional neural networks, internal transfer learning, small dataset.

I. INTRODUCTION

Nowadays automatically detecting and understanding the human actions in the video streams has become crucial in many applications such as intelligent video surveillance, auto-driving, somatic gaming and so on. This task is highly challenging when taking both accuracy and robustness into consideration. Considerable works are devoted to this topic in the human action recognition area. However, most of these methods highly rely on the reliable handcrafted features which consumes lots of time, and those features may vary with different datasets. Schuldt *et al.* [1] used support vector machines (SVM) in combination with several local spatial-temporal features as the recognition method. Scovanner *et al.* [2] introduced the 3D SIFT descriptor, based on which they extended the bag of words paradigm to videos to improve the action classification performance. Besides, other effective feature descriptors have also been utilized in the task of action recognition such as HOF (Histogram of Optical Flow) + MBH (Motion Boundary Histogram) [3], HOG (Histogram of Oriented Gradients) + HOF + BOF (Bag of Features) [4], DT (Dense Trajectories) + BOF [5] and so on.

In the recent years, Convolutional Neural Networks (CNN), one of the popular deep learning models, have shown great success in many computer vision tasks like image recognition [6], [7], image segmentation [8], object tracking [9], image super-resolution [10] and so on. The CNN tends to learn a hierarchy of features from low-level to high-level and researchers find that these features automatically learnt by the CNN are usually better than those handcrafted ones [11]. In human action recognition, researchers have put much efforts to build neural networks capable of capturing spatial-temporal features. Inspired by the 2D CNN, Ji *et al.* [12] developed a novel 3D CNN architecture which learnt from several adjacent video frames to obtain useful features. However, their model still took some pre-acquired low-level features (gradients and optical flows) as part of the input. After that, Karpathy *et al.* [13] proposed a multiresolution

CNN architecture combining two separate streams for large scale video classification. Encouraged by this work, many joint-training CNN models with multiple parallel networks were put forward [14]–[16] in different video recognition tasks and they indeed signficantly increased the classification accuracy. Later, Zhu *et al.* [17] presented a multimodal gesture recognition method combining the 3D CNN and the long-short-term-memory (LSTM) network. However, these deep learning models with complex architectures require considerable amount of training data [18]. With small sample size, they tend to suffer from the overfitting problem and fail to achieve good results.

In this paper, we are interested in the human action recognition in the dataset with small sample size such as KTH [1], Weizmann [19], UCF Sports [20] and VIVA challenge dataset [21]. To solve this problem, we apply the proposed the internal transfer learning algorithm to the 3DCNN for classification and achieve competitive results on these datasets.

II. METHODOLOGY

A. 3D CONVOLUTION NEURAL NETWORKS

In a typical 2D CNN, convolution operations are only applied to spatial dimensions. This conventional manner of convolution cannot capture the temporal features which are useful for action recognition. Different from 2D convolutions, 3D convolutions span the convolution operations over both spatial and temporal dimensions by convolving 3D kernels on the given video volumes. Involving the third dimension with such a 3D kernels allows to extract the useful spatial-temporal motion information that is crucial for action recognition.

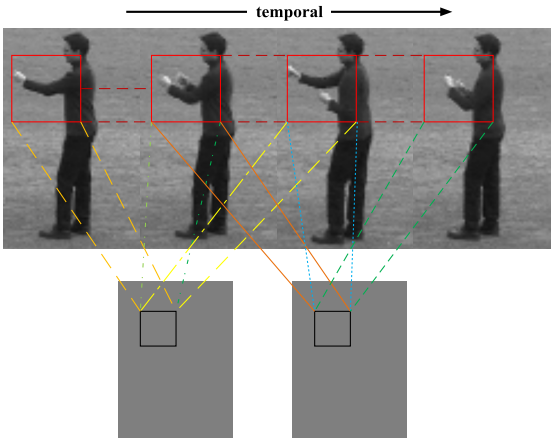


FIGURE 1. Illustration of the 3D convolutions in both spatial and time dimensions. In this figure two kernels are used and the size in the temporal dimension is 3.

Suppose we collect a few contiguous frames from the input videos and then stack them to form a data cube of $w \times h \times d$, with w, h and d representing the width, height and depth (temporal length) respectively. A 3D convolutional kernel of size $w' \times h' \times d'$ is then applied across this volume to generate a feature map. As illustrated in Fig 1, two sets of 3D kernels are used, and the output value v_{xyz} corresponding to the position (x, y, z) in the volume can be calculated by:

$$v_{xyz} = f\Big(\sum_{i=0}^{w'-1}\sum_{j=0}^{h'-1}\sum_{k=0}^{d'-1} w_{ijk} k_{(x+i)(y+j)(z+k)} + b\Big), \qquad (1)$$

where w_{ijk} is the weight at position (i, j, k) of the kernel, $k_{(x+i)(y+j)(z+k)}$ is the intensity of the image at position $(x+i, y+j, z+k)$ of the volume, f is the activation function and b is the bias entry. Thus, applying a given number of 3D kernels the convolutional network consequentially generates the same number of feature maps.

B. INTERNAL TRANSFER LEARNING

In this section, we present our proposed internal transfer learning (ITL) algorithm on small datasets which is a combination of transfer learning and the sub-data classification method. Transfer learning is an effective strategy for the cases with limited resources including training data and computation overhead. It aims to get the already learnt knowledge from a related task in the same domain and then apply it to the new task at hand. However, this kind of knowledge transfer requires other big datasets for pre-training and this extra condition cannot be always met. To address this problem, when dealing with an N-class classification task, we propose to divide this task to several binary ones, similar to the One-vs-One algorithm. However, unlike the time-consuming One-vs-One strategy, a significant difference is that we do not design $\frac{N\times(N-1)}{2}$ binary classifiers and sum up the classification results of all these classifiers at validation stage.

Specifically, for neural networks, we firstly design an N-class network architecture and train it on the whole dataset. From the validation results, we pick out several best-distinguished and worst-distinguished pairs of classes and then form an N'-class ($N' \leq N$) sub-dataset. Usually the general knowledge useful for classification can be learnt from the best-distinguished pairs on condition that the pairs are not quite similar. And the worst-distinguished pairs have the possibility of generating key classification information if dug into deeply. Thus we take both of them into consideration and set the proportion of them to 1:1 in the sub-dataset by experiments. In the following we use the same network to build $\frac{N'\times(N'-1)}{2}$ binary classifiers and pick out 5 models with the top-5 validation accuracy after rounds of training. These binary models are much easier to train and next we can utilize the learnt knowledge from these 5 pre-trained binary models and apply it to the original N-class classification network like the normal transfer learning procedure does: loading the weights and fine-tuning the last few newly-added layers. Finally the modified N-class model with the best performance is chosen as the best model. The details of the ITL algorithm is illustrated in Fig 2. The ITL algorithm makes use of the knowledge learnt from the sub-dataset and performs well in our later experiments.

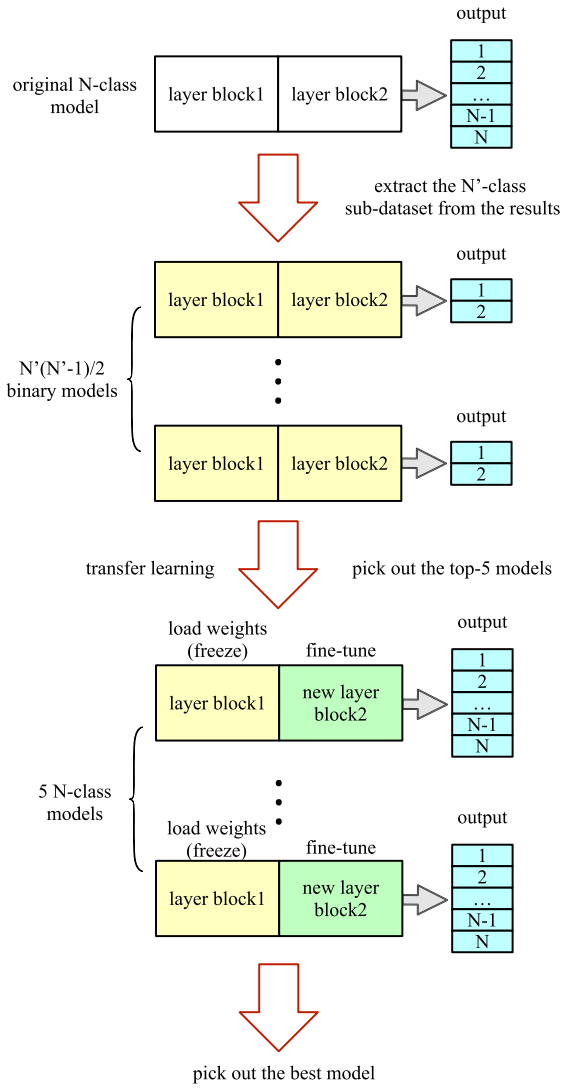


FIGURE 2. The implementation details of the internal transfer learning algorithm.

C. THE PROPOSED 3DCNN ARCHITECTURE

As mentioned above, our model is implemented by applying the ITL algorithm to the 3DCNN architecture. In the following, the descriptions are based on the experiments on the KTH dataset. At first we design a 6-class classification 3DCNN architecture inspired by AlexNet [6] which is depicted in Fig 3. This architecture consists of six layers of which the first five are 3D convolutional or max-pooling layers and the last one is a fully connected layer. The network receives video volumes of size $35 \times 55 \times 16$ as the input. For shorthand notation, we denote this model architecture by $C(16, 5, 5, 3) - S(2, 2, 1) - C(32, 6, 7, 3) - S(3, 3, 1) - C(64, 4, 7, 1) - F(1024)$, where $C(n, w, h, d)$ represents a convolutional layer with n kernels of size $w \times h \times d$, $S(w, h, d)$ represents a pooling layer with sub-sampling size of $w \times h \times d$ and $F(n)$ represents a fully-connected layer with n neurons. Finally the network outputs 6 values which is as same as the action types, representing the probability of each motion hypothesis with the help of the softmax regression function.

After the training of this 3DCNN completes, we apply the ITL algorithm by setting $N' = 4$ and pick out the top 5 binary models whose architecture is based on the 6-class 3DCNN but the number of output values is changed to 2. Then we keep all the layers remained except the last fully connected layer in the 3DCNN model and load the weights of the 5 pre-trained binary models by turn which are kept fixed in the following training procedure. After adding three new fully connected layers to the end of the former part, the redesigned ITL-3DCNN architecture are formed. As shown in Fig 3, the last three new fully connected layers output 512, 256 and 128 values separately. In the next stage, the ITL-3DCNNs are trained and the one with the highest validation accuracy is chosen as the best multi-class classifier.

D. TRAINING

To train the network, we choose the average cross-entropy as the loss function to minimize it:

$$l = -\frac{1}{N}\sum_{i=1}^{N} P(x^i) \cdot log(Q(x^i)), \tag{2}$$

where N is the total number of the samples of the data, x^i denotes the ith sample of the dataset, P and Q denote respectively the inherent probability distribution and the probability distribution of x predicted by the model.

During training, we adopt the momentum method when updating the weight parameter w_i using stochastic gradient descent with mini-batches of 50 samples. At each iteration, the weights are updated by the following rules:

$$v_{i+1} = \mu \cdot v_i - lr \cdot \nabla g, \tag{3}$$

$$w_{i+1} = w_i + v_{i+1}, \tag{4}$$

where i denotes the iteration index, μ denotes the momentum coefficient, v denotes the current velocity vector, lr denotes the learning rate and ∇g denotes the average value of the gradients with respect to w_i over the mini-batch at each iteration. We also bring up an update rule of the learning rate lr which proves to play a key role in the training procedure. The update rule is:

$$lr = lr \cdot \frac{1}{1 + d \cdot i}, \tag{5}$$

where d is the decay parameter and i is the iteration index. The decay of learning rate leads to smaller training errors and a better generalization capability. Specifically, in our fine-tuning procedure, the parameters μ and d are set to 0.9 and 0.008 separately after abundant experiments.

In our experiments the weights in each layer are initialized from a truncated normal distribution centered on 0 with standard deviation $std = \sqrt{\frac{2}{n}}$ where n denotes the input or output connections at a layer according to [22]. And we choose the ReLU activation function and set the biases for all the layers to 0 by the same reason. During the training stage, we apply drop-out strategy [23] with probability 0.5 after all the fully-connected layers and L2 regularization on the

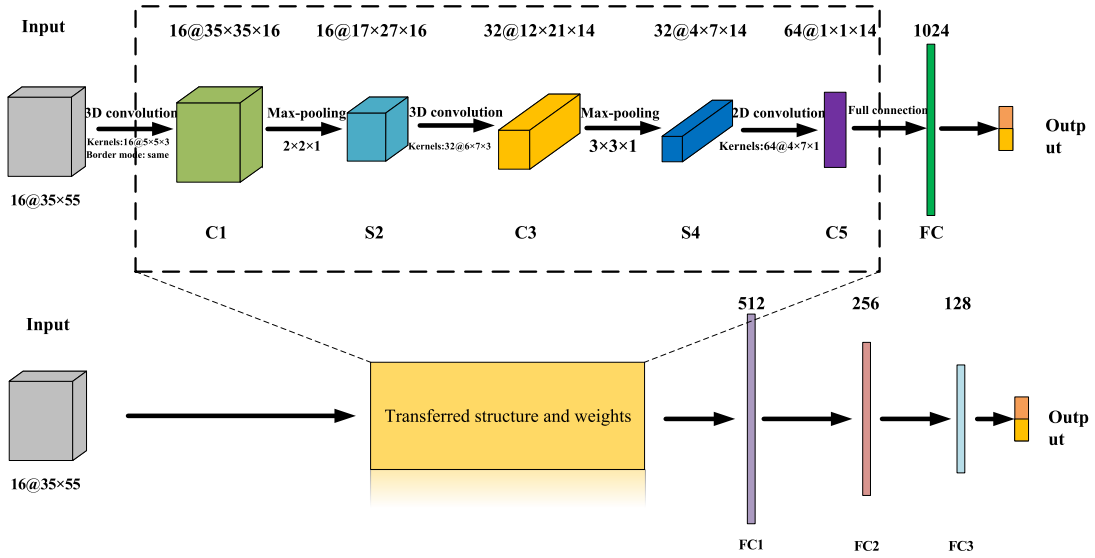


FIGURE 3. The 3DCNN and the redesigned ITL-3DCNN architectures. In the architectures, *C* stands for the convolutional layer, *S* stands for the pooling layer and *F* stands for the fully-connected layer. In our work, 3DCNN is firstly used for multi-class classification and then modified for binary classification. Finally, the multi-class ITL-3DCNN utilizes part of the well-trained binary 3DCNN's structure and weights.

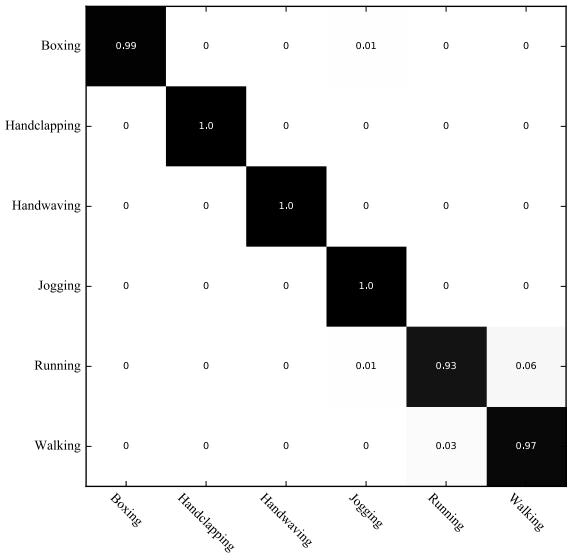


FIGURE 4. Confusion matrix for the KTH dataset. Total accuracy 98.2%.

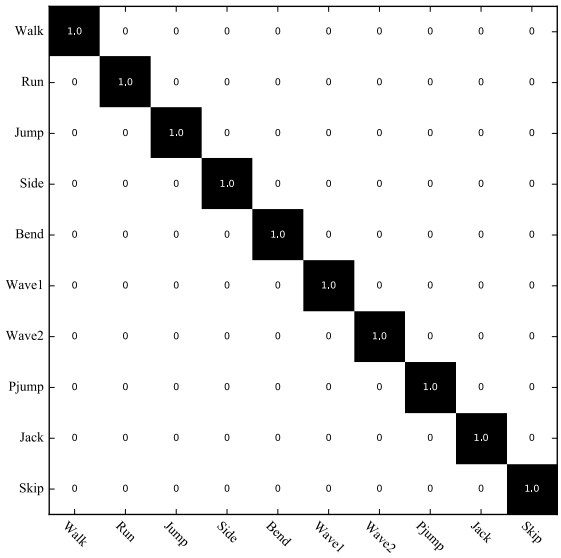


FIGURE 5. Confusion matrix for the Weizmann dataset. Total accuracy 100%.

weights of all the layers to overcome the possible overfitting problem. To accelerate the training, we also apply batch normalization [24] to the response of each layer.

III. RESULTS AND DISCUSSION

A. KTH DATASET AND WEIZMANN DATASET

On the KTH and Weizmann dataset, we perform our experiments using the proposed 3DCNN architecture in Fig 3 and achieve promising results.

The KTH dataset contains six different types of human actions performed by 25 people in four different scenarios. And each action has 100 video samples. The Weizmann dataset consists of 90 video clips which can be divided into 10 action classes. The original video volumes of the two datasets are firstly down-sampled to the size of $40 \times 60 \times 16$. Then we apply the data augmentation strategy that we randomly crop out 19 times of clips with size $35 \times 55 \times 16$ from the video volumes on the training data which accounts for 90% of the dataset. And we use 10-fold cross-validation when evaluating the performance of the model.

We set $N' = 4$ on both the KTH and Weizmann dataset. Then we have 6 binary 3DCNNs on the sub-dataset. Note that training errors on these binary models converge very fast so that the entire procedure is not very time consuming. Then the

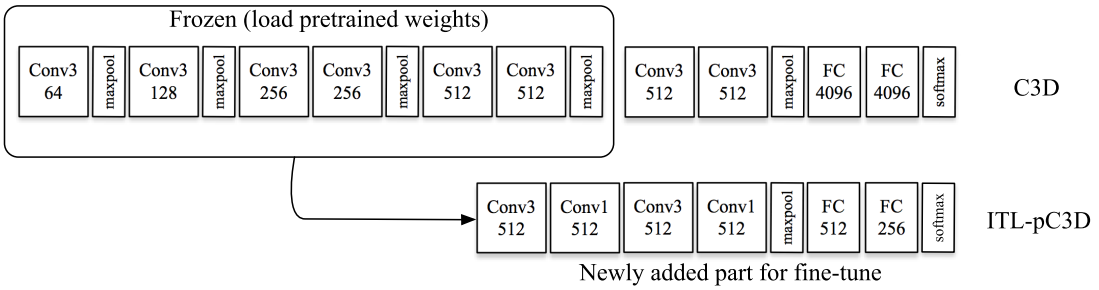


FIGURE 6. The C3D and ITL-pC3D architectures. In the convolution block, $ConvN_1N_2$ represents N_2 convolutional kernels of size $N_1 \times N_1 \times N_1$. And *FCN* represents *N* nodes in the fully-connected layer.

models with top 5 validation accuracy are picked out and their weights are fed into the redesigned ITL-3DCNN by turn for the next stage of training. Finally, after rounds of training, the two best multi-class classification models achieve the holistic recognition accuracy of 98.2% and 100% on the two datasets respectively. The confusion matrices are depicted in Fig 4 and Fig 5. The comparison with other peer works is reported in Table 1.

TABLE 1. Comparisons of our work to the state-of-art methods on the KTH and Weizmann dataset. The 3DCNN is the multi-class classification model trained from scratch using the proposed architecture.

Method	KTH	Method	Weizmann
Ji [12]	90.2	Moshe [25]	100
Wang [3]	94.2	Lena [26]	100
Liu [27]	95.5	-	-
Sadanand [28]	98.2	-	-
3DCNN	91.0	3DCNN	98.7
ITL-3DCNN	**98.2**	ITL-3DCNN	**100**

B. UCF SPORTS DATASET AND VIVA CHALLENGE DATASET

The UCF Sports dataset is a very challenging dataset which has 150 video clips from 10 actions. On one hand, the dataset has complex backgrounds and camera motion and clutter. One the other hand, its sample size is very small. For example, the skateboarding actions have only 6 videos and they vary a lot from each other, which brings great obstacle to the deep learning training procedure. The VIVA challenge dataset which has 19 hand gesture classes is also very challenging for its settings like cluttered background and volatile illumination.

The small sample size and complex data contents make it very difficult to obtain high recognition accuracy when training neural networks on the two datasets' raw input data from scratch. Especially on the VIVA challenge dataset, the state-of-art accuracy is only 77.5% achieved by a two-stream CNN architecture with complex data augmentation method in Molchanov *et al.*'s work [14]. To prove the effectiveness of ITL algorithm, we test the 3DCNN, ITL-3DCNN, C3D, ITL-C3D, pC3D, ITL-pC3D models on the two datasets and report the results in Table 2.

C3D is a 15-layer architecture for action recognition proposed by Tran *et al.* [29]. However, we can find that even this

TABLE 2. The performance comparison of different methods on the UCF-Sports and VIVA challenge datasets. In the table, the 3DCNN and the ITL-3DCNN use the same proposed architectures performed on the KTH dataset.

Method	UCF-Sports	Method	VIVA
Wang [30]	85.5	Omar [31]	58.7
Stephen [32]	91.3	Eshed [33]	64.5
Shao [34]	93.4	Molchanov [14]	77.5
Ghodrati [35]	95.7	-	-
3DCNN	78.1	3DCNN	74.5
ITL-3DCNN	83.9	ITL-3DCNN	76.3
C3D	81.1	C3D	81.0
ITL-C3D	84.4	ITL-C3D	84.3
pC3D	91.6	pC3D	93.9
ITL-pC3D	**93.6**	ITL-pC3D	**96.1**

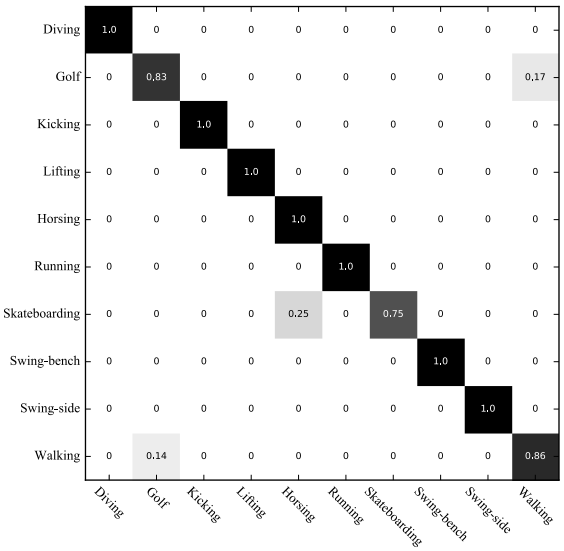


FIGURE 7. Confusion matrix for the UCF-Sports dataset. Total accuracy 93.6%.

highly effective model cannot achieve a high accuracy faced with the two datasets. When ITL is added, the accuracies increase. Going further, we incorporate the ITL algorithm into the pre-trained C3D model (pC3D) and achieve satisfactory results. As shown in Fig 6, the upper part network is the C3D network loaded with pre-trained weights on the Sports-1M dataset [13]. Then we remove several bottom layers and add a few new layers to the former part whose

weights remain fixed. Next we put this new architecture into the binary classification training procedure and use the top-5 models' weights as initialization for the final multi-class classification by turn. Finally, by this way, our ITL-pC3D achieve 93.6% and 96.1% validation accuracy on UCF-Sports and VIVA challenge dataset. 2/3 of the data is used for training and the remained for validation and 10-fold cross-validation is adopted. The confusion matrices are depicted in Fig 7 and Fig 8.

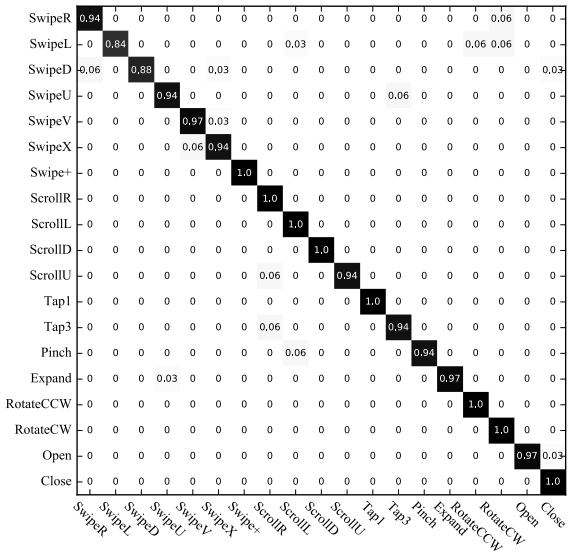


FIGURE 8. Confusion matrix for the VIVA challenge dataset. Total accuracy 96.1%.

Our model is trained on one NVIDIA GTX1080 8GB GPU and each experiment is trained for roughly 8 hours. The experiments on the four datasets prove the effectiveness of the ITL algorithm. We can find that ITL can always improve the original model's performance. On the challenging UCF Sports and VIVA dataset, it can be noticed that simple networks like the proposed 3DCNN or complex networks like C3D can hardly obtain high recognition accuracy if trained from scratch on the raw input data. But when the ITL algorithm is applied to the pre-trained network, we can achieve a satisfactory high validation accuracy.

IV. CONCLUSIONS

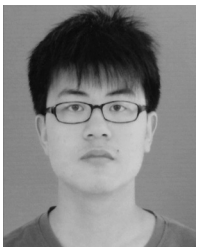
In this paper, we focus on the human action recognition problem. Instead of using conventional ways to capture the handcrafted features, we utilize the 3D convolutional neural network to automatically extract useful spatial-temporal features. To overcome the difficulty of training with datasets of small sample size, we propose the internal transfer learning algorithm (ITL). Our method achieves competitive results on several datasets compared to the peer works.

REFERENCES

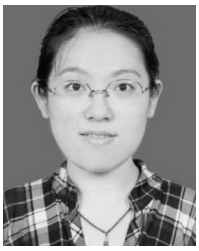
[1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. IEEE 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3. Aug. 2004, pp. 32–36.
[2] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th ACM Int. Conf. Multimedia*, 2007, pp. 357–360.
[3] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3551–3558.
[4] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.
[5] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3169–3176.
[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[7] H. M. Bui, M. Lech, E. Cheng, K. Neville, and I. S. Burnett, "Object recognition using deep convolutional features transformed by a recursive network structure," *IEEE Access*, vol. 4, pp. 10059–10066, 2016.
[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 91–99.
[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[10] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1637–1645.
[11] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image Vis. Comput.*, vol. 60, pp. 4–21, Apr. 2017.
[12] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1725–1732.
[14] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 1–7.
[15] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2983–2991.
[16] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. (2015). "Towards good practices for very deep two-stream ConvNets." [Online]. Available: https://arxiv.org/abs/1507.02159
[17] G. Zhu, L. Zhang, P. Shen, and J. Song, "Multimodal gesture recognition using 3-D convolution and convolutional LSTM," *IEEE Access*, vol. 5, pp. 4517–4524, 2017.
[18] X.-W. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014.
[19] S.-F. Wong and R. Cipolla, "Extracting spatiotemporal interest points using global information," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.
[20] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.
[21] S. Hadfield and R. Bowden, "Hollywood 3D: Recognizing actions in 3D natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3398–3405.
[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[24] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167
[25] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2005, pp. 1395–1402.
[26] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[27] L. Liu, L. Shao, and P. Rockett, "Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition," *Pattern Recognit.*, vol. 46, no. 7, pp. 1810–1818, 2013.
[28] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1234–1241.
[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.
[30] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, 2012.
[31] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 716–723.
[32] S. O'Hara and B. A. Draper, "Scalable action recognition with a subspace forest," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1210–1217.
[33] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2368–2377, Dec. 2014.
[34] L. Shao, X. Zhen, D. Tao, and X. Li, "Spatio-temporal Laplacian pyramid coding for action recognition," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 817–827, Jun. 2014.
[35] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. Van Gool, "DeepProposals: Hunting objects and actions by cascading deep convolutional layers," *Int. J. Comput. Vis.*, vol. 124, no. 2, pp. 115–131, 2017.

TIAN WANG (S'13–M'16) received the M.S. degree from Xi'an Jiaotong University, China, in 2010, and the Ph.D. degree from the University of Technology of Troyes, France, in 2014. He is an Assistant Professor with the School of Automation of Science and Electrical Engineering, Beihang University. His research interests include computer vision and pattern recognition.



YANG CHEN received the B.S. degree in automation from Beihang University in 2017. His academic interests include image processing, video surveillance, and machine learning.



MENGYI ZHANG received the M.S. and Ph.D. degrees from the University of Reims Champagne-Ardenne, France, in 2011 and 2016, respectively. She is an Assistant Professor with the College of Electrical Engineering and Control Science, Nanjing Tech University, China. Her research interests include wireless sensor networks and topological signal processing.

JIE CHEN (S'12–M'14) received the B.S. degree in information and telecommunication engineering from Xi'an Jiaotong University, Xi' an, China, in 2006, the Dipl.-Ing. degree in information and telecommunication engineering from the University of Technology of Troyes (UTT), Troyes, France, the M.S. degree in information and telecommunication engineering from Xi'an Jiaotong University in 2009, and the Ph.D. degree in systems optimization and security from UTT in 2013.

From 2013 to 2014, he was with the Lagrange Laboratory, University of Nice Sophia Antipolis, France. From 2014 to 2015, he was with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. He has been a Professor with the Center of Intelligent Acoustics and Immersive Communications, Northwestern Polytechnical University, since 2015. His current research interests include adaptive signal processing, distributed optimization with applications in hyperspectral image analysis, acoustic signal processing, and bioinformatics.

Dr. Chen has been recognized with the Thousand Talents Plan (Youth Program) Award in China. He was the Technical Co-Chair of IWAENC'16 held in Xi'an.

HICHEM SNOUSSI received the Diploma degree from Ecole Supérieure d'Electricité in 2000, and the DEA and Ph.D. degrees from the University of Paris-Sud in 2000 and 2003, respectively. From 2003 to 2004, he was a Post-Doctoral Researcher with the Institut de Recherche en Communications et Cybernétiques de Nantes. Since 2010, he has been a Full Professor with the University of Technology of Troyes. His research interests include signal processing, computer vision, and machine learning.